

- *in this portion:*
  - Requirements
  - Philosophy
  - Overview of Online System
    - Hardware architecture
    - Software architecture
- *presented by Fritz Bartlett:*
  - Controls and Monitoring
  - Test Beam efforts
- *and then back to me:*
  - ...other software components
  - Schedules and manpower

# Requirements

- *Functionality:*
  - Control of detector
    - hardware settings
    - configuration
    - triggering
  - Monitoring of
    - hardware state
    - trigger operations
    - data flow
    - data content

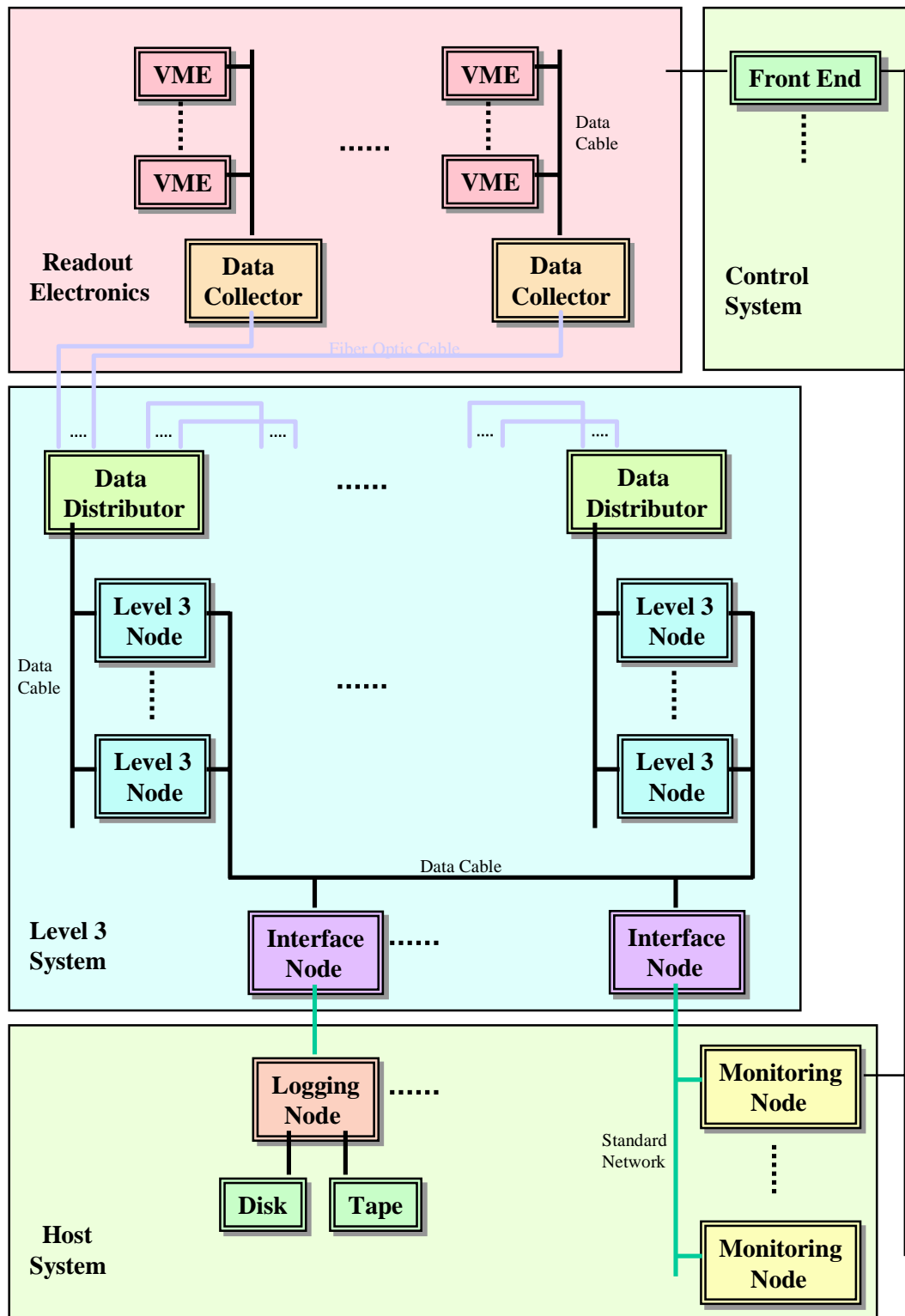
## Requirements (cont'd)

- *Bandwidth:*
  - 20 Hz @ 250 Kbytes/event  
*5 Mbytes/sec*
  - Upgrade path to 50 Hz  
*12.5 Mbytes/sec*
  - Burst rate of 100 Hz (local)  
*25 Mbytes/sec*
- These figures set the scale for the total Run II data volume. The targets for the Level 1 / Level 2 / Level 3 trigger rates are 10 kHz / 1 kHz / 20 Hz. The DAQ system needs to be capable of up to 50 Hz if the Level 3 rejection rate cannot be met. It might be expected that the full capacity will be used, putting a strain on offline resources. In the end, the DAQ rate may be tailored to match offline capacity.

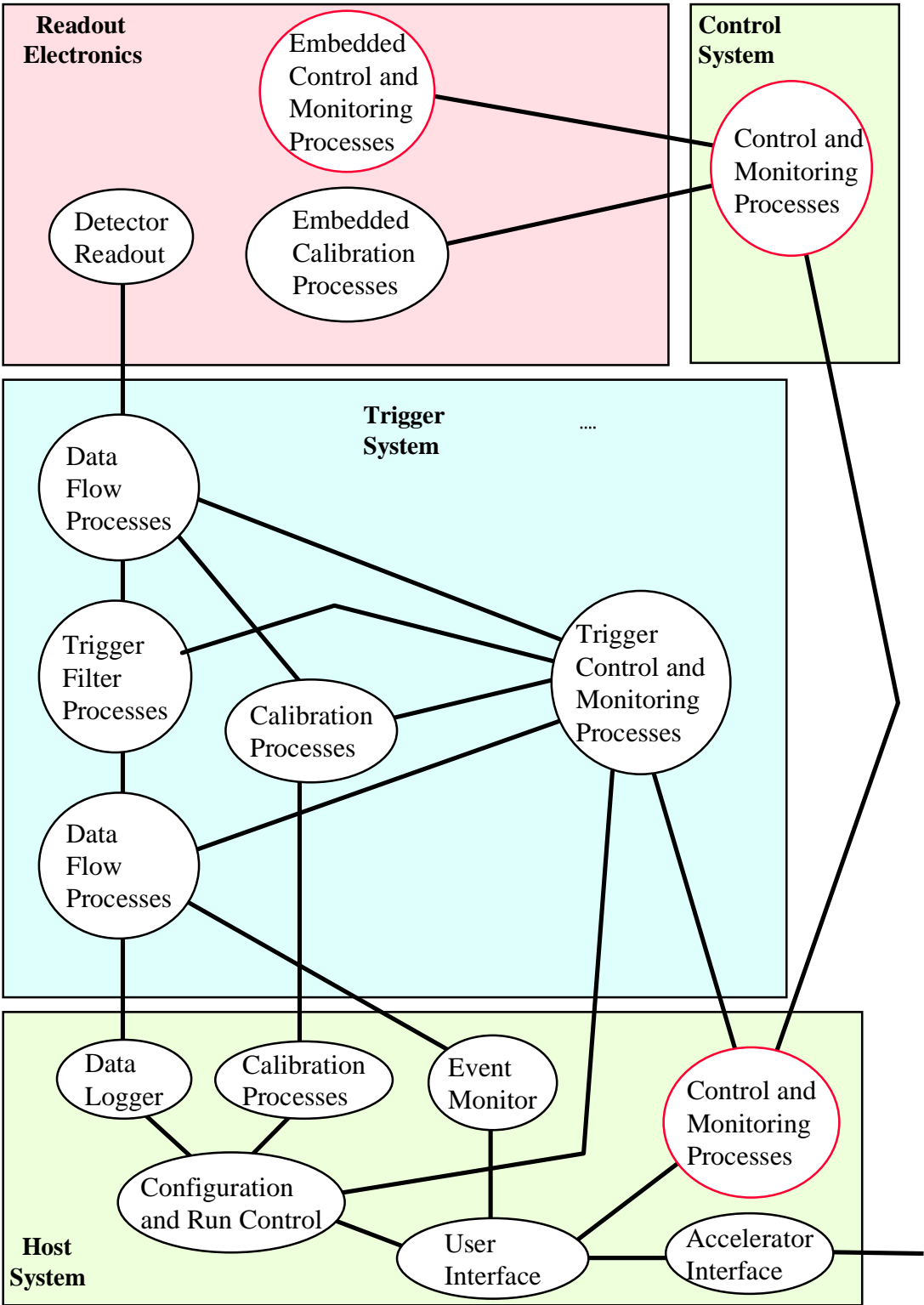
# Philosophy

- *Learn from Run I*
  - Similar architecture
- *Beg, borrow, and steal...*
  - EPICS, DART
- *Follow the crowd...*
  - UNIX host system
  - NT filter and control nodes
  - VXWORKS front ends
- *Stay with the rest of DØ*
  - C++ (though will allow C, FORTRAN as needed)
  - DØ Code Management, Graphics, Databases, ...

# Overview: Hardware



## Overview: Software



## Part 3 introduction...

- *Run I Legacy*
  - items that we have to live with...
- *Online tools development*
  - software upon which many applications are built
- *Component list*
  - the highlights only...
- *Schedule and Manpower*
  - the highlights and summary...

## Run I Legacy

- *Control system components:*
  - Token Ring network
  - Shea/Goodwin 68K processors
  - 1553, Vertical Interconnect
  - DØ High Voltage system
- > Mixed Ethernet and Token Ring network; mixed Front End systems; dual personality control system
  
- *Data Path:*
  - Level 3 infrastructure and architecture
- > Distributed processors with associated communication and control requirements



## Tool Development

- *Inter-task communication*
  - Decision to use upgraded version of Run I Client/Server package
- *Control & Monitoring*
  - Evaluating EPICS in Test Beam; will integrate with Run I CDAQ
  - Basis for:
    - Downloading
    - Alarms
- *Event Distribution, Data Logging*
  - Evaluating DART in Test Beam

## Component List

- *Control & Monitoring Applications:*
  - Hardware Database
  - Front End management
  - Gateway
  - High Voltage
  - Cryo, Gas control
  - Parameter Page
  - Clock Server
  - Alarm system
  - Data Logger
- *System Performance Monitoring*
- *Accelerator interface*
- *Configuration and Run Control Applications:*
  - “COOR”
  - User interface
- *Data collection and monitoring:*
  - Data logger
  - Tape manager
  - Trigger monitor
- *Calibration:*
  - in Front End
  - in Level 3
  - at Host
- *Event Monitoring:*
  - Event distribution
  - “EXAMINE”

## Schedule

- *Milestones:*

- 1997

Mar	Test Beam operations
Jul	COOR - trigger system protocols
Jul	Version 1 Inter-task communication
Jul	1/6 <sup>th</sup> hardware purchased
Oct	Control & Monitor download path
Oct	Level 1 Framework installation / minimal trigger

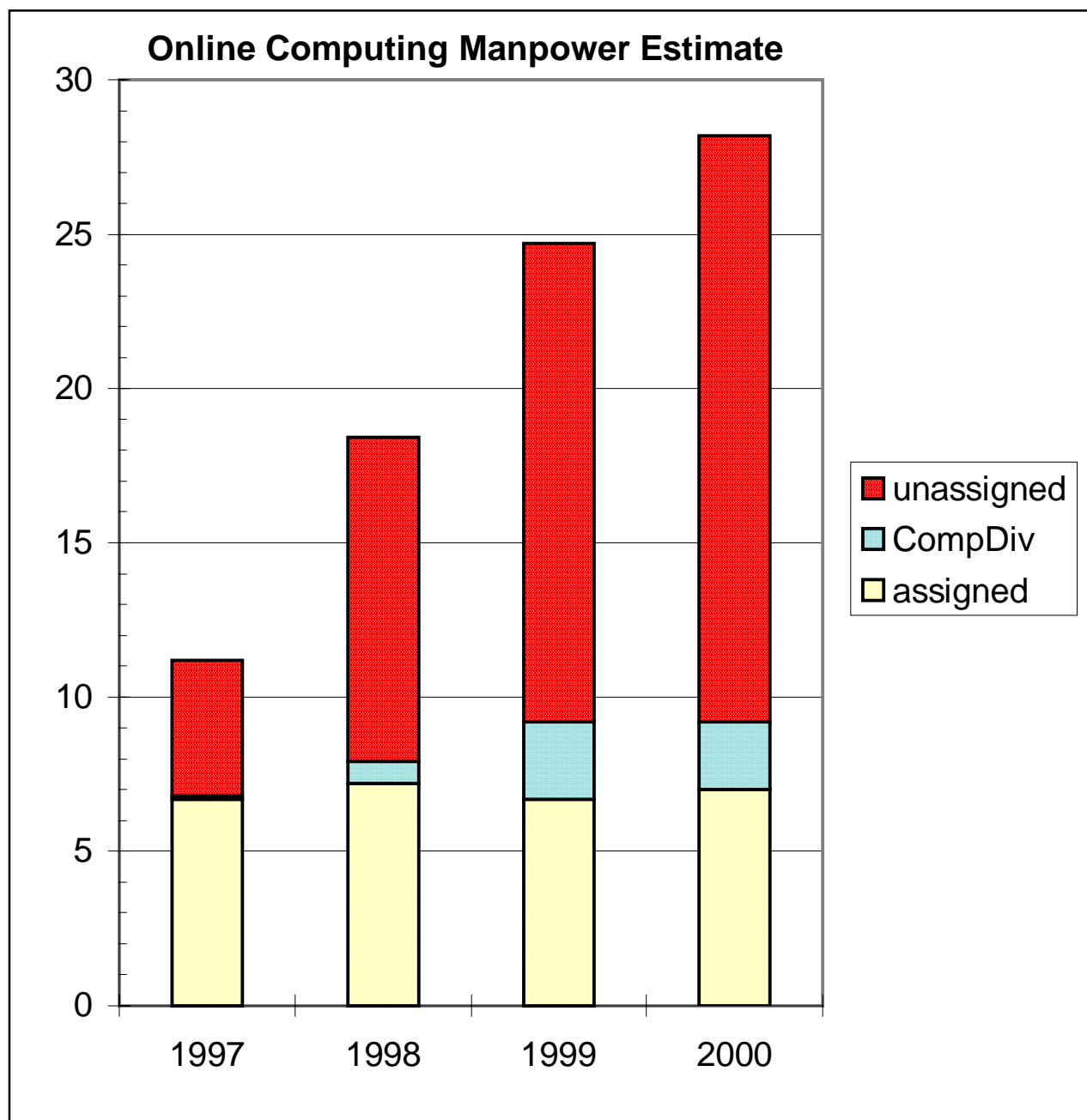
- 1998

Mar	COOR - Level 2 protocols
Jun	Readout VME to Level 3
Jun	2/6 <sup>th</sup> hardware purchased
Aug	Level 2 commissioning / extended trigger

- 1999

Jan	Readout VME to Host
Jun	Active triggers
Jun	3/6 <sup>th</sup> hardware purchased

# Manpower Profile



## Why not \_\_\_\_ ?

- Why not VMS?
  - Lab as whole moving away; diminishing expertise
  - Restricted source of tools - EPICS, DART
- Why not FORTRAN?
  - Very little of core of Run I system was FORTRAN - it was mostly PASCAL
  - Not inherently a structured language; would need vendor extensions or would need to move to FORTRAN 90; the latter considered as drastic as C++
  - Hard to find young people interested in FORTRAN